Pathlib

```
from os import chdir
from pathlib import Path


def main() -> None:
    # current working directory and home directory
    cwd = Path.cwd()
    home = Path.home()
    print(f"Current working directory: {cwd}")
    print(f"Home directory: {home}")

    # creating paths
    path = Path("/usr/bin/python3")

    # using backslashes on Windows
    path = Path(r"C:\Windows\System32\cmd.exe")

    # using forward slash operator
    path = Path("/usr") / "bin" / "python3"

    # using joinpath
    path = Path("/usr").joinpath("bin", "python3")

    # reading a file from a path
    path = Path.cwd() / "settings.yaml"
    with path.open() as file:
        print(file.read())

    # reading a file from a path using read_text
    print(path.read_text())

    # resolving a path
    path = Path("settings.yaml")
    print(path)
    full_path = path.resolve()
```

```
print(full_path)

# path member variables
print(f"Path: {full_path}")
print(f"Parent: {full_path.parent}")
print(f"Grandparent: {full_path.parent.parent}")
print(f"Name: {full_path.name}")
print(f"Stem: {full_path.stem}")
print(f"Suffix: {full_path.suffix}")

# testing whether a path is a directory or a file
print(f"Is directory: {full_path.is_dir()}")
print(f"Is file: {full_path.is_file()}")

# testing whether a path exists
print(f"Full path exists: {full_path.exists()}")
wrong_path = Path("/usr/does/not/exist")
print(f"Wrong path exists: {wrong_path.exists()}")

# creating a file
new_file = Path.cwd() / "new_file.txt"
new_file.touch()

# writing to a file
new_file.write_text("Hello World!")

# deleting a file
new_file.unlink()

# creating a directory
new_dir = Path.cwd() / "new_dir"
new_dir.mkdir()

# changing to the new directory
chdir(new_dir)
print(f"Current working directory: {Path.cwd()}")

# deleting a directory
new_dir.rmdir()
```

```
if __name__ == "__main__":
    main()
```

Revision #1
Created 6 March 2024 02:10:31 by victor
Updated 7 March 2024 01:49:41 by victor